

Optimización de recursos con algoritmos de balanceo de carga



índice

- Introducción
- Solución implementada
- Escenarios de prueba
- Estructura de ficheros creados
- Algoritmos implementados
 - Random
 - Round Robin
 - IP Random
- Conclusiones
- Posibles mejoras

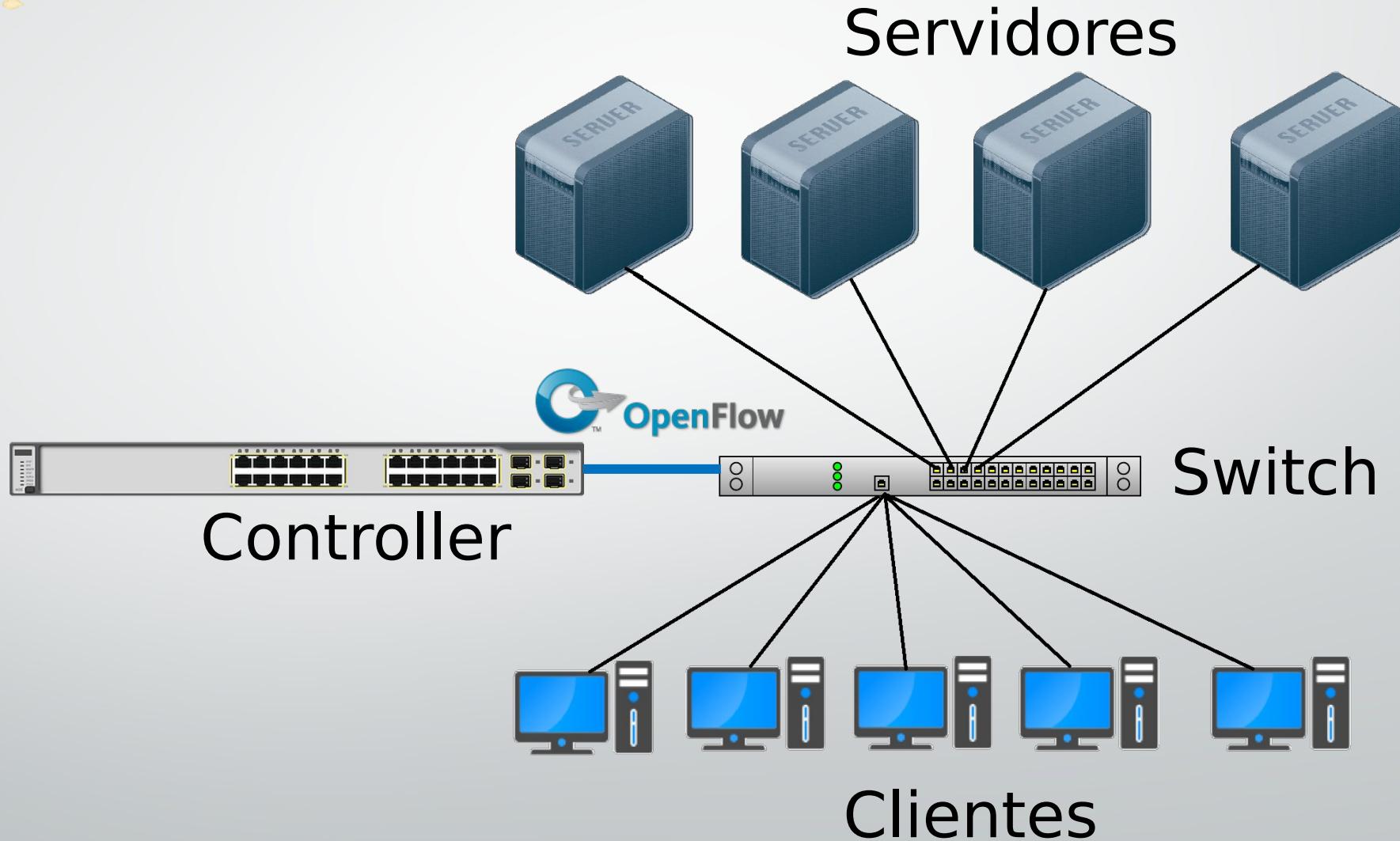
1. Introducción

Situación actual: Empresa de hosting web sufre saturación en sus servidores debido al creciente número de usuarios.

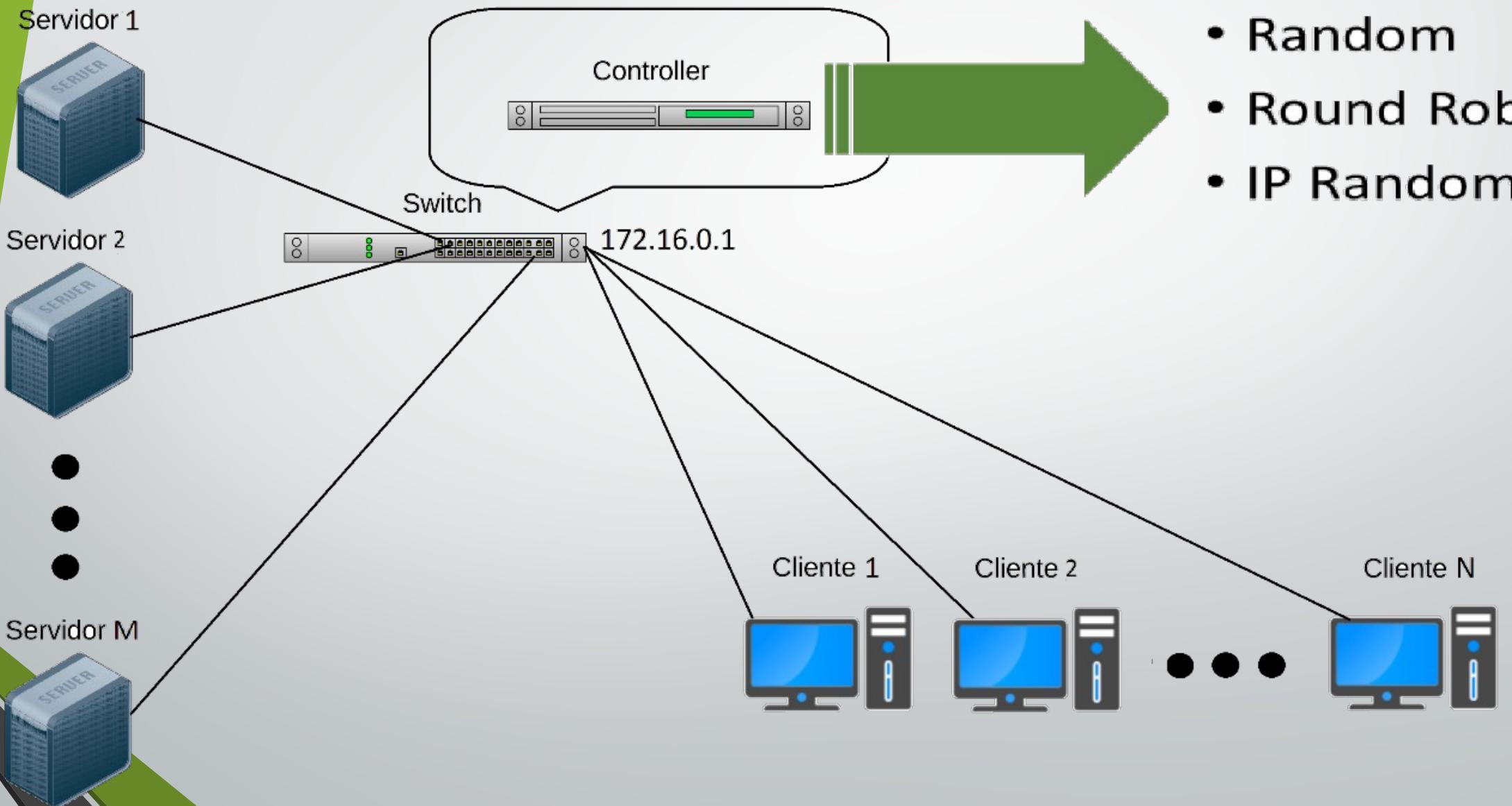
Se va a diseñar y simular un sistema de balanceo de carga para obtener una mayor eficiencia y rendimiento, mejorando así la situación de la empresa.



2. Solución implementada

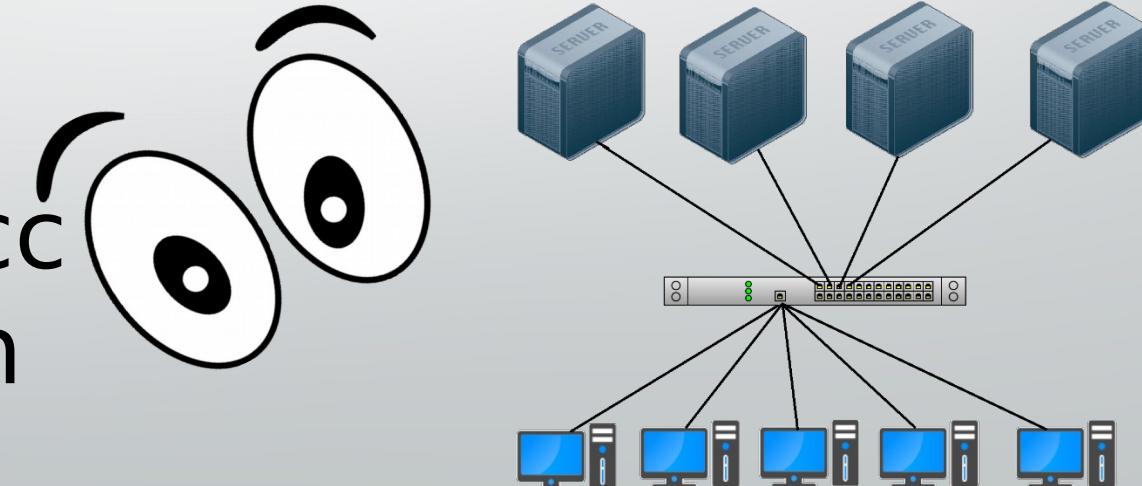


3. Escenario de prueba

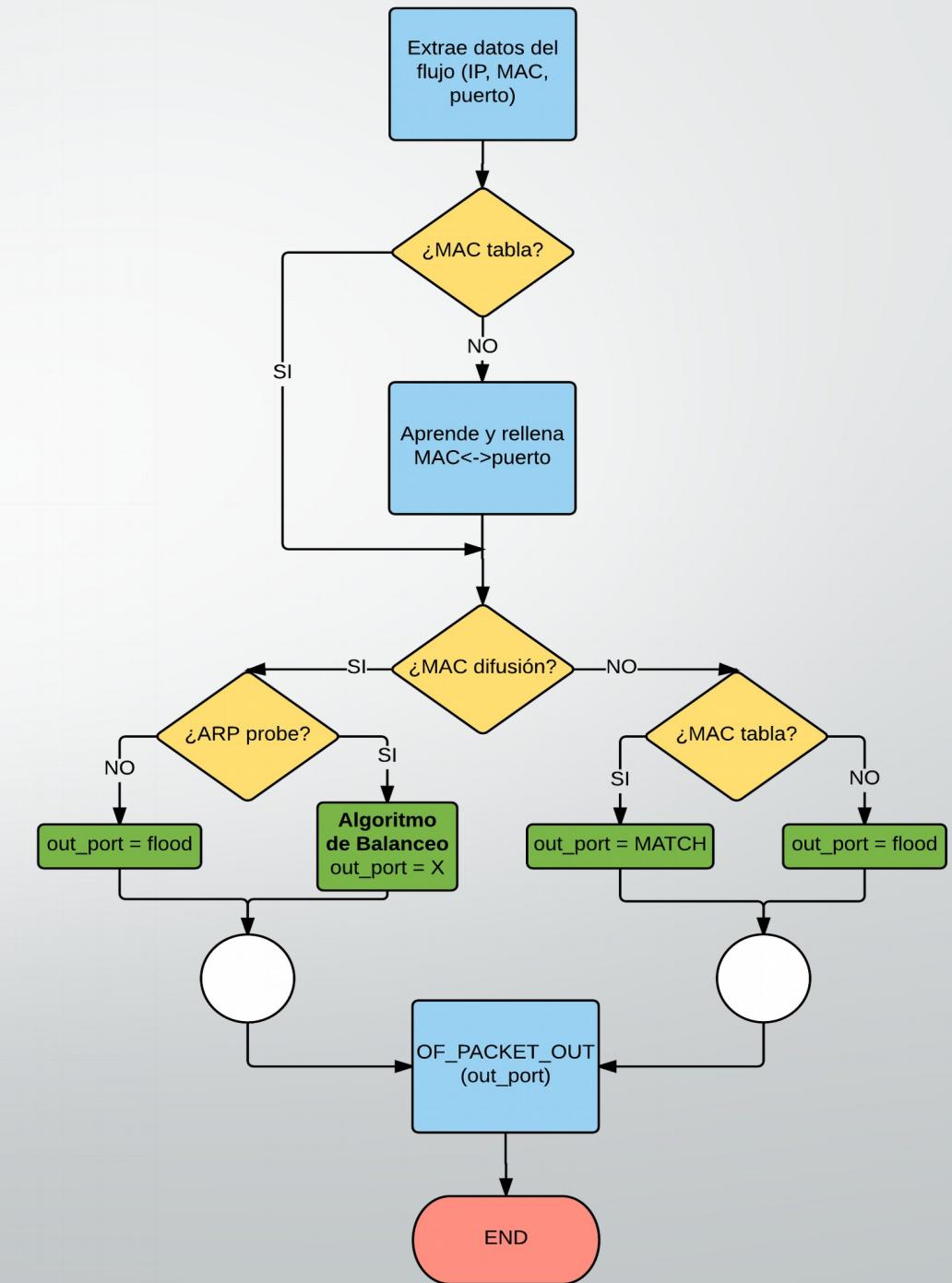
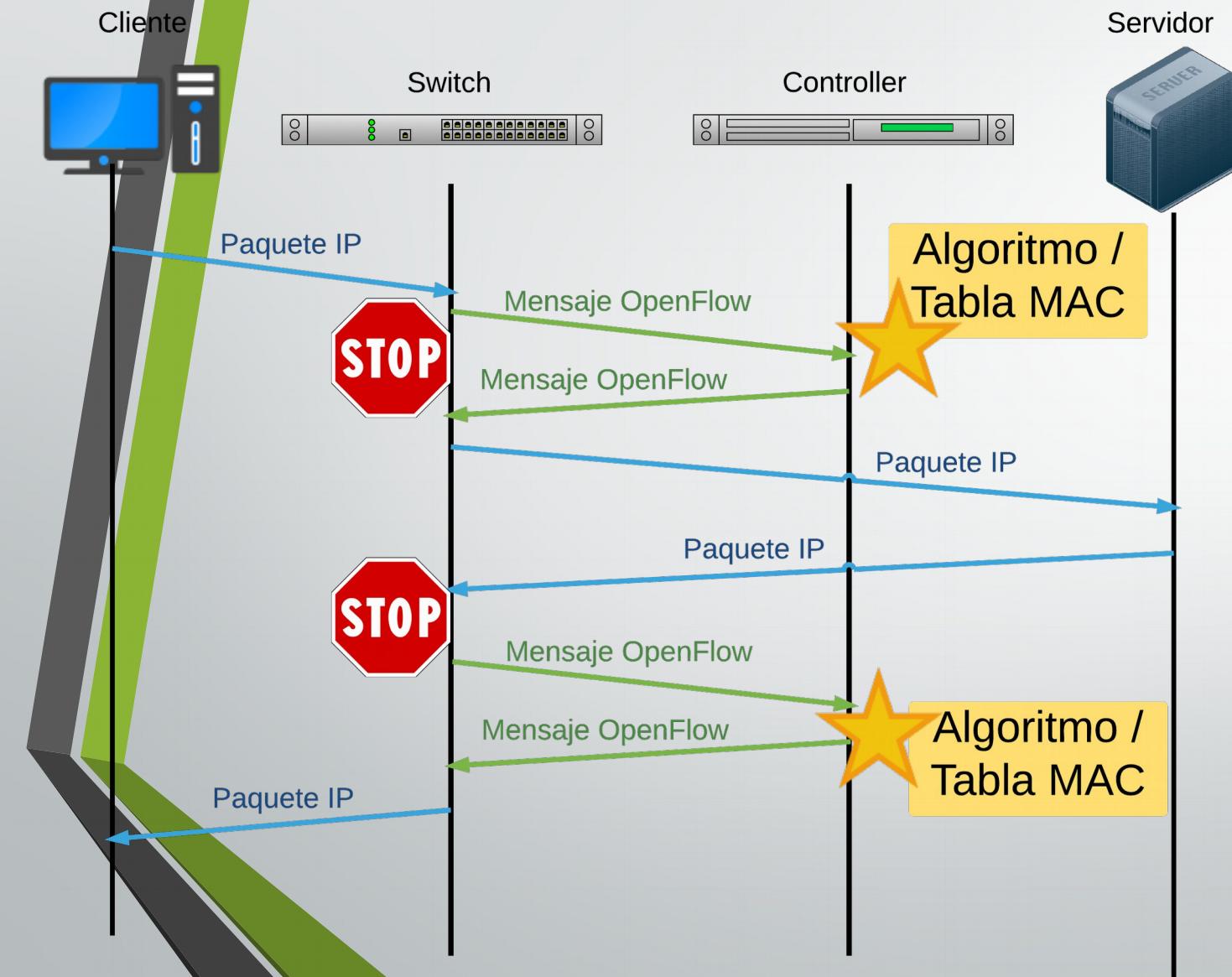


4. Estructura de ficheros creados

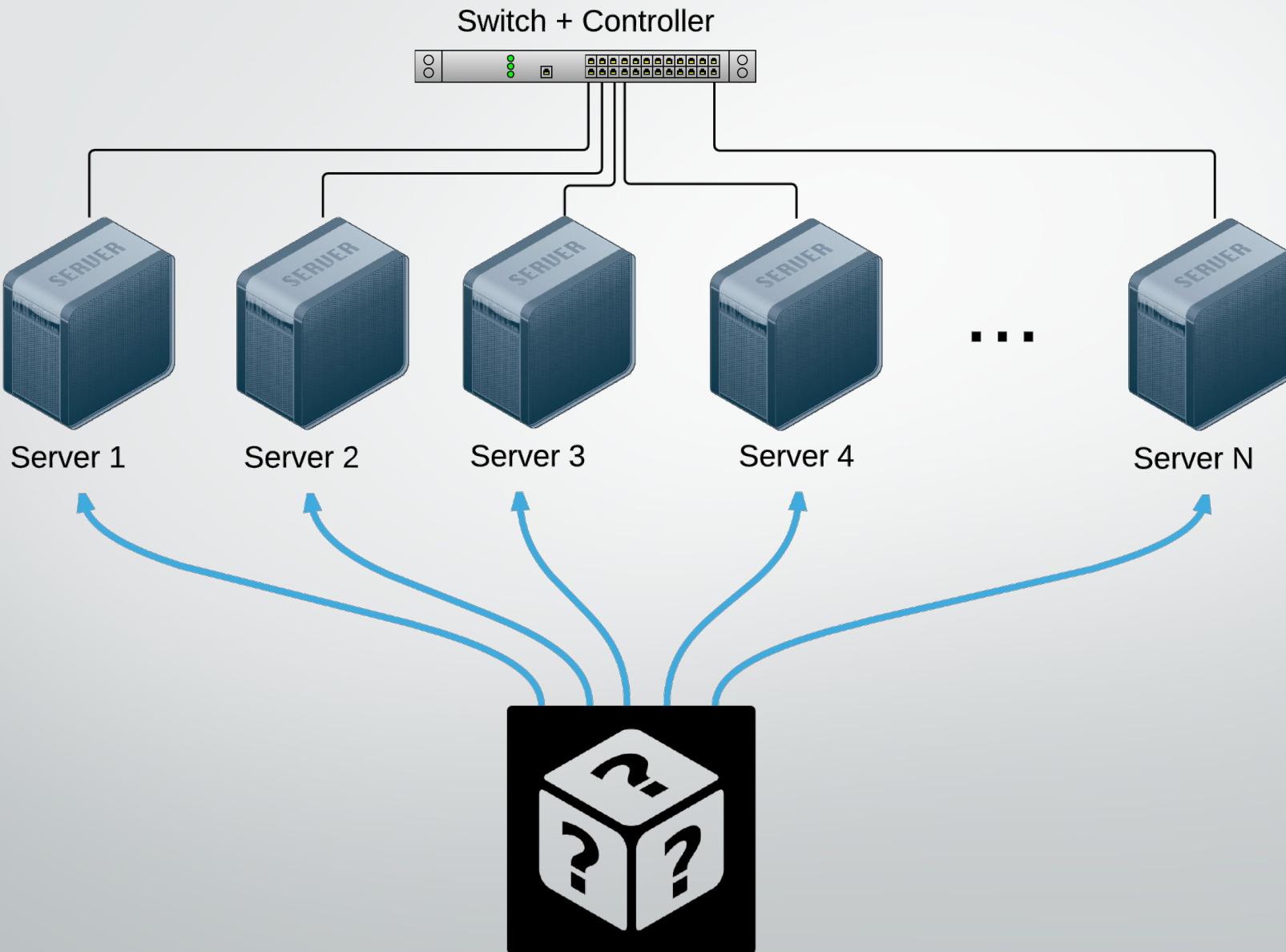
- **Escenario y Simulación:**
 - Topología.cc
 - Loadbalancer.h
- **Controllers:**
 - Controller.h
 - Random_controller.cc
 - Roundrobin_controller.cc
 - Iprandomcontroller.cc
- **Obtención de datos:**
 - Observador.cc
 - Observador.h



5. Algoritmos

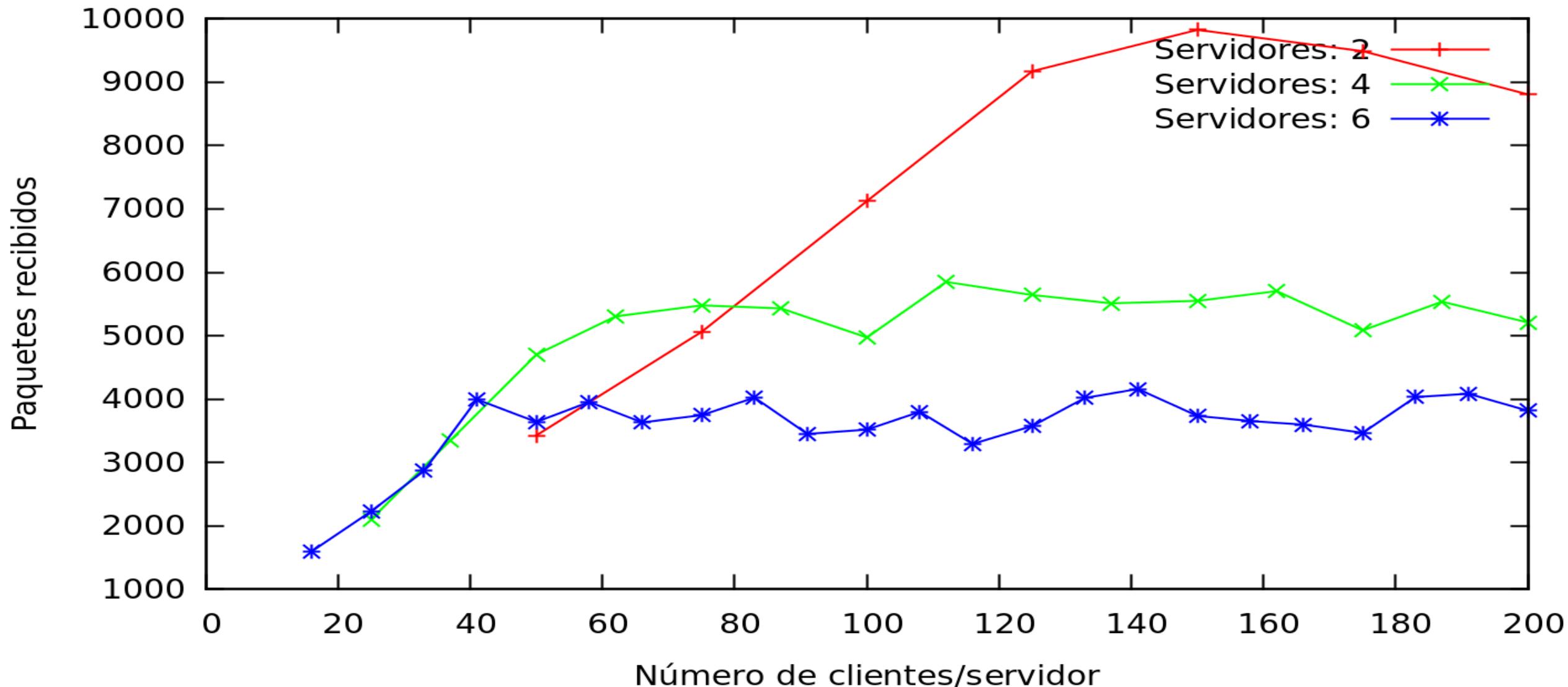


5.1 Algoritmo Random



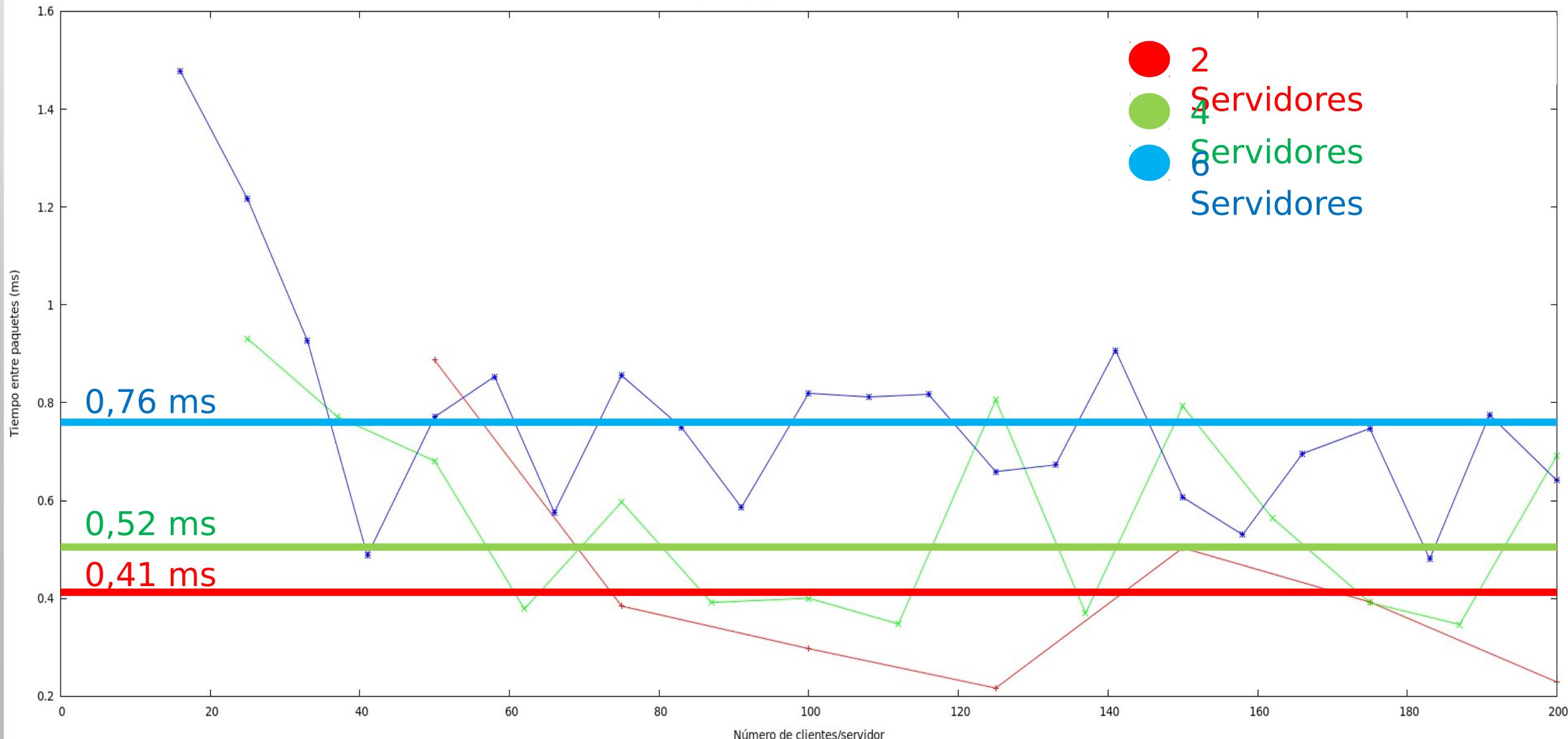
5.1 Algoritmo Random

Gráfica 1: Paquetes recibidos aumentando el nº clientes en random

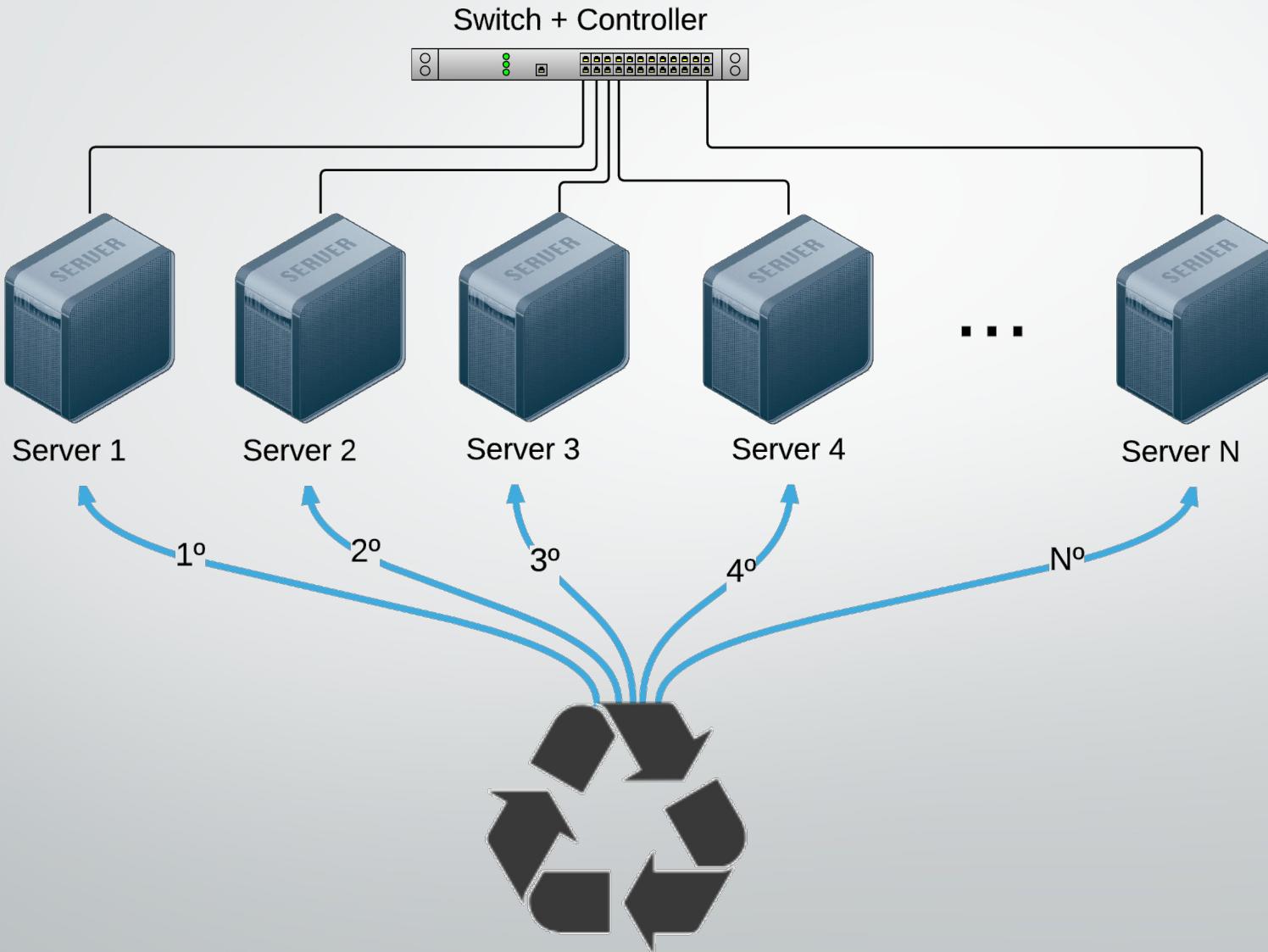


5.1 Algoritmo Random

Gráfica 2: Tiempo entre paquetes aumentando el nº clientes en random

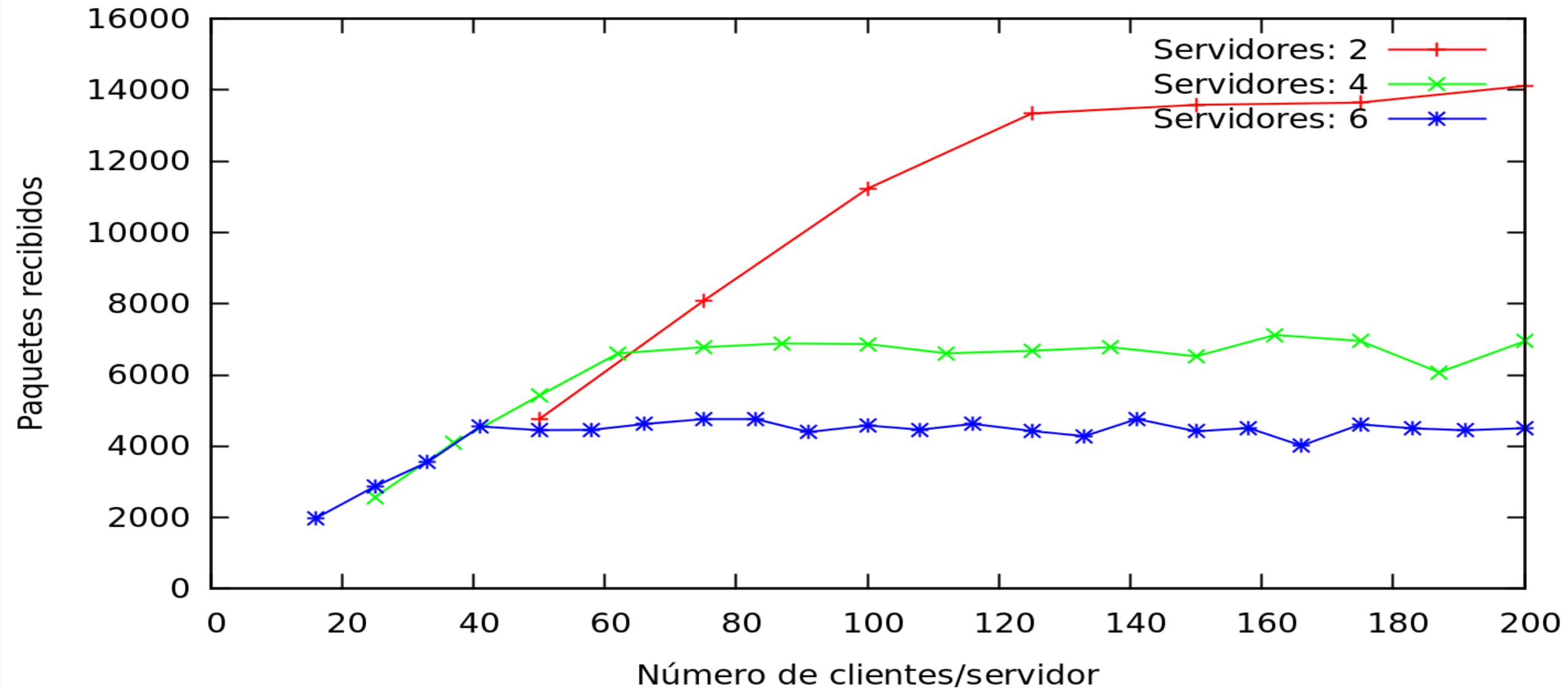


5.2 Algoritmo Round Robin



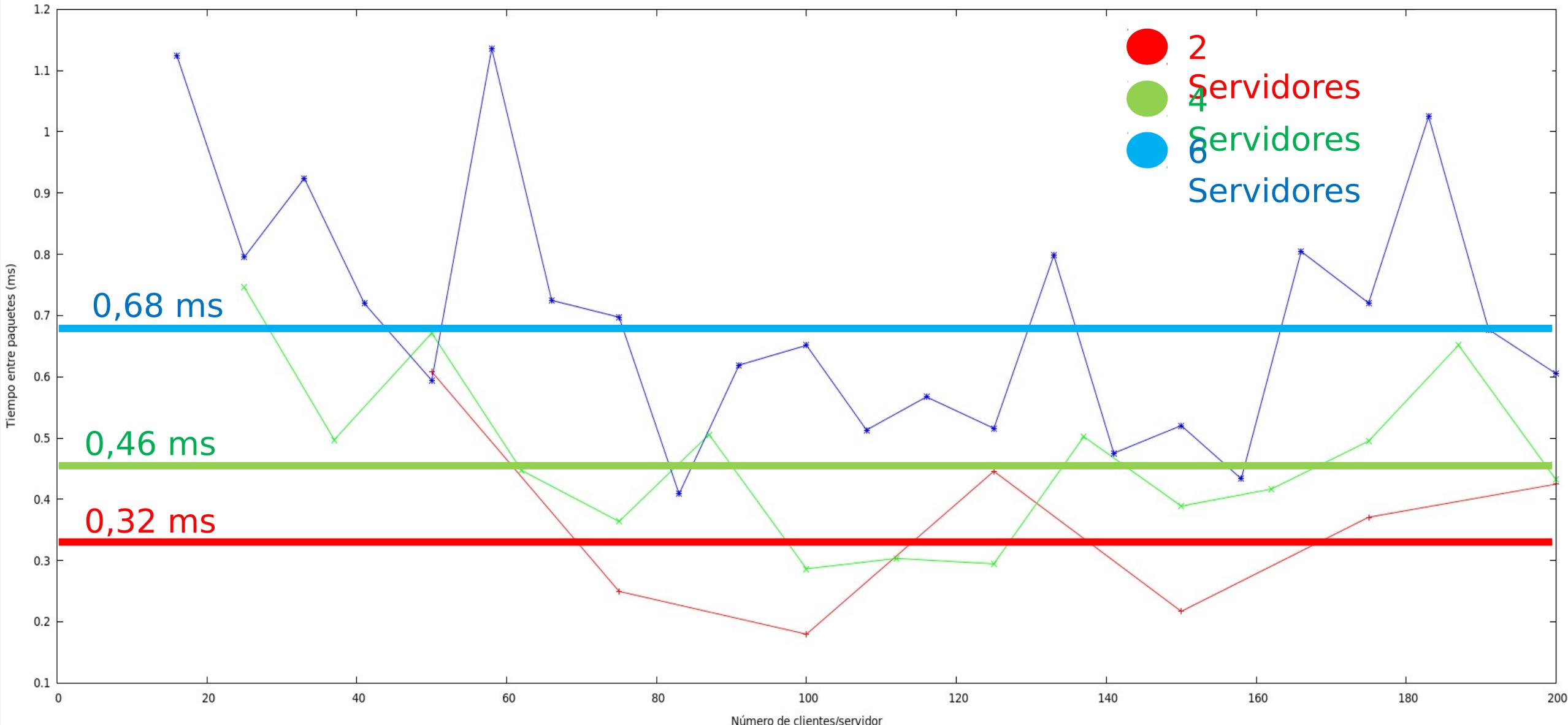
5.2 Algoritmo Round Robin

Gráfica 1: Paquetes recibidos aumentando el nº clientes en round-robin

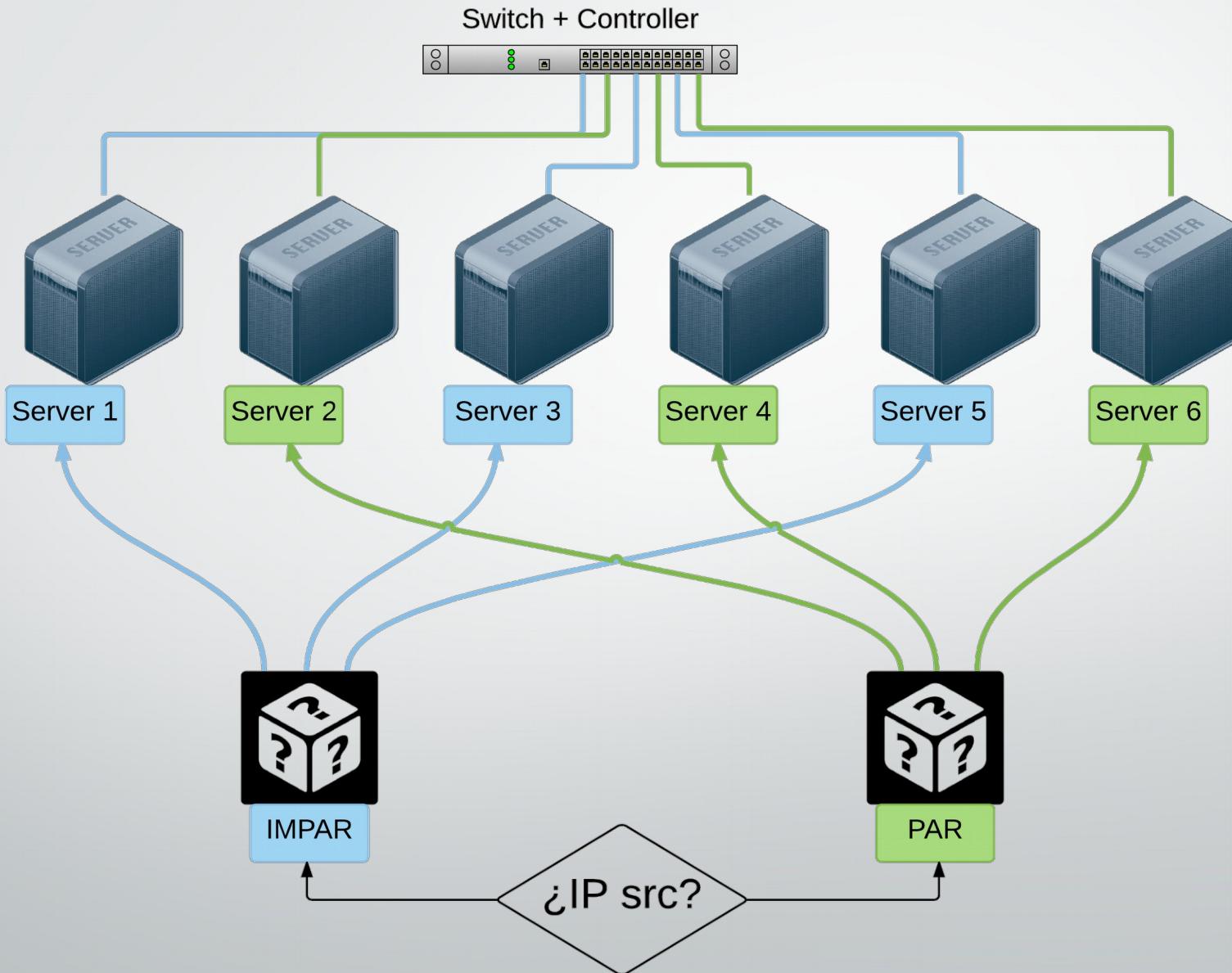


5.2 Algoritmo Round Robin

Gráfica 2: Tiempo entre paquetes aumentando el nº clientes en round-robin

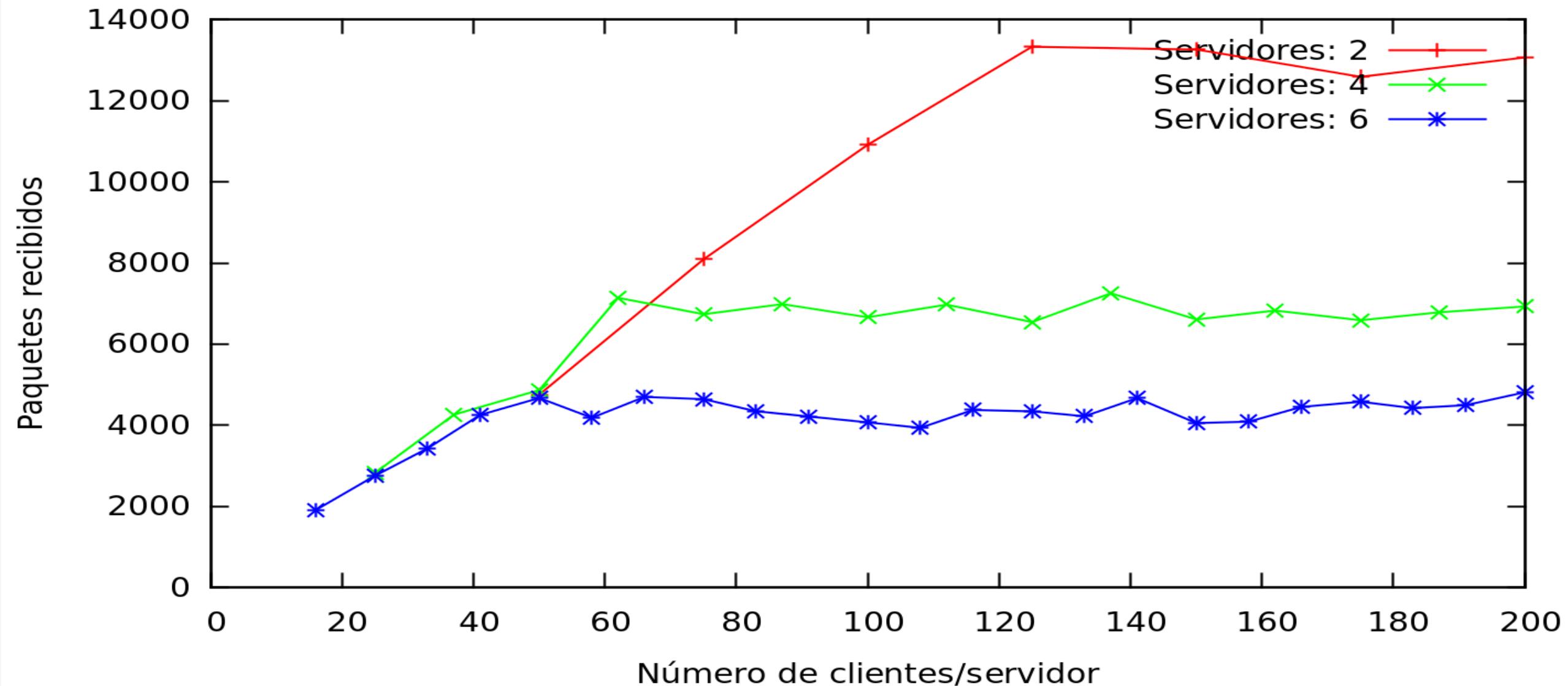


5.3 Algoritmo IP Random

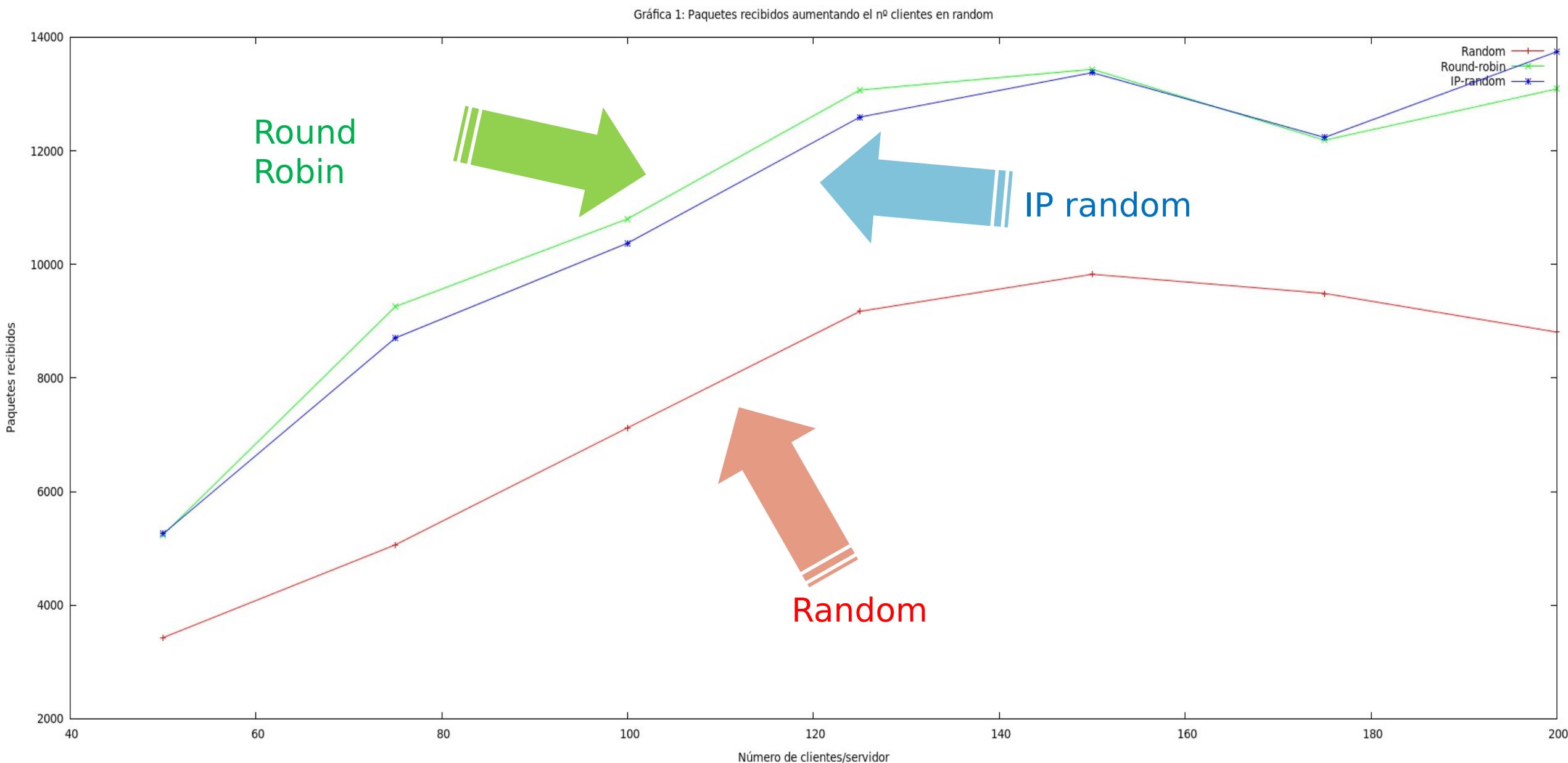


5.3 Algoritmo IP Random

Gráfica 1: Paquetes recibidos aumentando el nº clientes en IP-random



5.4 Comparativa



6. Conclusiones

- Como se ha podido observar en la diapositiva anterior:
 - El algoritmo ***Random*** es el que **peor rendimiento** ofrece
 - *Round-Robin* e *IP Random* ofrecen un rendimiento similar, siendo **mejor *Round-Robin*** cuando la **relación** clientes por servidor es **inferior** a **150**, en cambio, para una **relación** cliente por servidor **superior** a **175** es **mejor *IP Random***.
- Justificación:
 - Ante un número **elevado** de **clientes** para los mismos servidores *IP Random* ofrece mayor rendimiento debido a que se garantiza que las **peticiones** de los clientes sean atendidas siempre por un **número** de **servidores** más **acotado**.
 - *Random* es la peor opción en toda la casuística debido a que no hay **nada** de **predicción** en el algoritmo.

7. Posibles mejoras

- Nuevos algoritmos de balanceo
- Optimizar manejo de errores
- Algoritmos adaptativos en función del tráfico (QoS)
- Realizar el balanceo a nivel 3

FIN

